Assignment 3: kNN and SVM

UVA CS 4501 :
Machine Learning (Fall 2018)

Out: Oct. 7 2018
Due: Oct. 21, Sun midnight 11:59pm, 2018 @ Collab

a *The assignment should be submitted in the PDF format through Collob. If you prefer hand-writing QA parts of answers, please convert them (e.g., by scanning or using PhoneApps like camscanner) into PDF form.*

b *For questions and clarifications, please post on piazza.*

c *Policy on collaboration:*

Homework should be done individually: each student must hand in their own answers. It is acceptable, however, for students to collaborate in figuring out answers and helping each other solve the problems. We will be assuming that, with the honor code, you will be taking the responsibility to make sure you personally understand the solution to any work arising from such collaboration.

d *Policy on late homework: Homework is worth full credit at the midnight on the due date. Each student has three extension days to be used at his or her own discretion throughout the entire course. Your grades would be discounted by 15% per day when you use these 3 late days. You could use the 3 days in whatever combination you like. For example, all 3 days on 1 assignment (for a maximum grade of 55%) or 1 each day over 3 assignments (for a maximum grade of 85% on each). After you've used all 3 days, you cannot get credit for anything turned in late.*

1 KNN and Model Selection (K) (programming)

- Purpose 1: To implement a very simple classifier, k-nearest neighbors (KNN), from scratch.
- Purpose 2: To implement a 4-folds Cross Validation strategy for selecting the best K for KNN classification. Please feel free to reuse the cross-validation codes you had in the previous HWs.

We will uses euclidean distance as the measurement of closeness in KNN. We will test your implementation on the *"Movie_Review_Data.txt"*

- Please use the "knn.py" template to guide your work. Please follow the instructions and the function names/descriptions in the template. Feel free to cross-check your implementation against sci-kit's KNN. Other requirements or recommendations are the same as HW1 and HW2.
- 2.1 Implement the read_csv method. The last column of the data file includes a 0 or 1 label. Please feel free to reuse your codes from HW1 or HW2.

  Att: there are many ways to read in the reference datasets, e.g., our template reads in the whole file and put it into one numpy array. (This is differently from our ridge regression template in which our template actually read the file into two numpy array, one for Xval, the other for Yval. Both ways are correct.)

  Att: please remember to shuffle the whole data before performing the CV.
- 2.2 Implement the CV method:

$$training, testing = fold(data, i, kfold = 4)$$

Where "data" is the full data you have loaded, "i" is the iteration of cross validation, and "kfold" is the total number of folds you specify for CV. This method will be as simple as splitting the data matrix into the training fold and testing fold for the current "i".

ATT: please feel free to try other folds of CV like kfold=3-fold or kfold=10-fold.

- 2.3 Implement the classify method:

$$predictions = classify(training, testing, K)$$

Where training includes the training set of data, testing is the testing set, and K is the number of data points that KNN takes into consideration when labeling an unlabeled data point. Note here the training data is part of the testing (classify) algorithm. In KNN, we label a new data point based on the k points that are the closest to it in our train dataset. In this function, for each testing point, please find its K nearest neighbor in the training set (those having the smallest Euclidian distance to the testing point). Then we label the testing point according to the majority voting rule.

- 2.4 Implement the calc_accuracy method:

$$acc = calc\_accuracy(predictions, labels)$$

Where predictions is the list of 0 or 1 predictions obtained from the classify method and labels is the true label for the testing points (the last column of the test data).

(Hint1: If your accuracy is below 50%. Please consider how the order of the samples are dictated by the class.)

- 2.5 Run the code with K = (3, 5, 7, 9, 11, 13). Report the accuracy and the best K. Discuss why some values work better than others.
- A bar graph is recommended to show the change of accuracy with K. By using K as x-axis and accuracy as y-axis
- Att: we will not consider the speed in grading your kNN codes.

2 Support Vector Machines with Scikit-Learn and preprocessing

- (1) Install the latest stable version of scikit-learn following directions available at http://scikit-learn.org/stable/install.html Also make sure to download "salary.labeled.csv" from collab.
- (2) For this assignment, you will create a program using scikit-learn's C-Support Vector Classifier: sklearn.svm.SVC ;
- Given a proper set of attributes, the program will be able to determine whether an individual makes more than 50,000 USD/year. You may use code from previous homework to help you import the data.
- Bear in mind you will need to do some preprocessing of the data before applying the SVM. See course slide for details.
- Two sample files are provided. The unlabeled sample set "salary.2Predict.csv" is a text file in the same format as the labeled dataset "salary.labeled.csv", except that its last column includes a fake field for class labels.

- 2.1 You are required to generate/ predict labels for samples in "salary.2Predict.csv".
- 2.2 We will evaluate your output 'predictions' - an array of strings (">50K" or "<=50K") according to the true labels of these test samples (ATT: you don't have these labels !!!). This simulates a Kaggle-competition in which test labels are always held out and only team-ranking will be released after all teams have submitted their predictions. When grading this assignment, we will rank all students' predictions. So please try to submit the best performing model that you can!
- 2.3 You need to report the CV classification accuracy results by performing 3-fold cross validation (CV) on the labeled set "salary.labeled.csv" using at least three different SVM kernels you pick. Please provide details about the kernels you have tried and their performance (e.g. 3CV classification accuracy) including results on both train and test folds into the writing. For instance, you can summarize the results into a table with each row containing kernel choice, kernel parameter, CV train accuracy and CV test accuracy.
- (Hint: you can choose SVM kernels like, basic linear kernel / polynomial kernel, varying its parameters / RBF kernel, varying its parameters).

Submission Instructions: You are required to submit the following :
(The starting code, 'income_classifier.py', has been provided in Collab.)

1. A python program that includes the statements:

   ```
   clf = SvmIncomeClassifier()
   trained_model, cv_score = clf.train_and_select_model(``salary.labeled.csv")
   ```

   It should be able to train and select a model using a set of hyperparameters on the training data, these hyperparameters can be hard coded.

   Next, we should be able to use the learned/ trained_model to classify samples in an unlabeled test set using the following function:

   ```
   predictions = clf.predict(``salary.2Predict.csv",trained_model)
   ```

2. A file "predictions.txt" generated by:

   ```
   clf.output_results(predictions)
   ```

   Please do not archive the file or change the file name for the automated grading.

3. A table in your PDF submission reporting classification accuracy (score) averaged over the test folds, along with details of the kernels, best performing hyperparameter C for each case etc.

Classes: >50K, <=50K.
Attributes:
age: continuous.
workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
fnlwgt: continuous.
education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
education-num: continuous.
marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
occupation: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
sex: Female, Male.
capital-gain: continuous.
capital-loss: continuous.
hours-per-week: continuous.
native-country: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.

Table 1: About the data in SVM coding.

3 Sample Exam Questions:

Each assignment covers a few sample exam questions to help you prepare for the midterm and the final. (Please do not bother by the information of points in some the exam questions.)

Question 1. Support Vector Machine

Soft-margin linear SVM can be formulated as the following constrained quadratic optimization problem:

$$argmin_{\{w,b\}} \frac{1}{2} w^T w + C \sum_{i=1}^{m} \epsilon_i$$

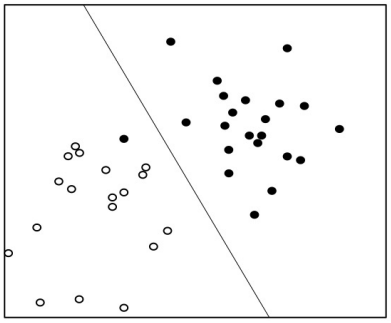
such that

$$y_i(w^T x_i + b) \geq 1 - \epsilon_i$$
$$\epsilon_i \geq 0 \qquad \forall i$$

where C is the regularization parameter, which balances the margin (smaller $w^T w$) and the penalty of mis-classification (smaller $\sum_{i=1}^{m} \epsilon_i$).

(a) (**True/False**) Number of support vectors do not depend on the selected value of C. Please provide a one-sentence justification.

In the following figure, $n_{C=0}$ is the number of support vectors for C getting close to 0 ($\lim_{C \to 0}$) and $n_{C=\infty}$ is the number of support vectors for $C = \infty$.



C=1

(b) Select the correct option:( )

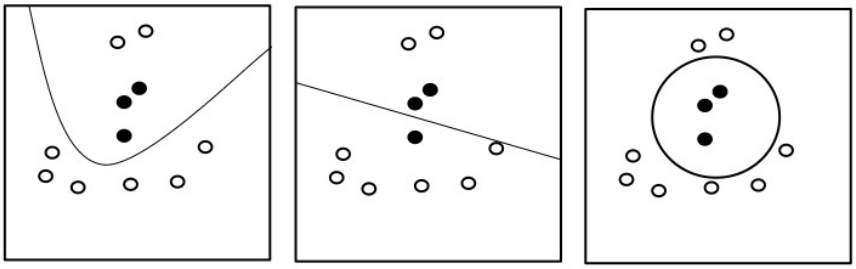
(1) $n_{C=0} > n_{C=\infty}$

(2) $n_{C=0} < n_{C=\infty}$

(3) $n_{C=0} = n_{C=\infty}$

(4) Not enough information provided

(c) Match the following list of kernels used for SVM classification with the following figures:



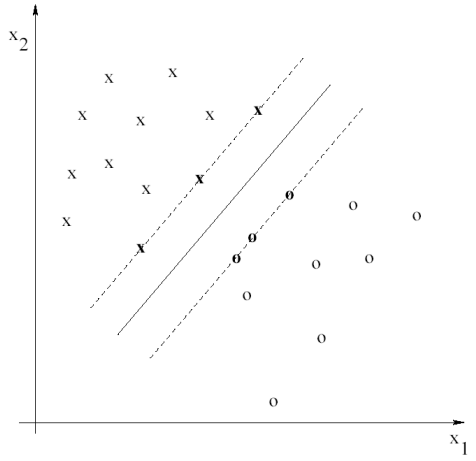
_____ _____ _____

(1) Linear Kernel : $K(x, x') = x^T x'$

(2) Polynomial Kernel (order = 2) : $K(x, x') = (1 + x^T x')^2$

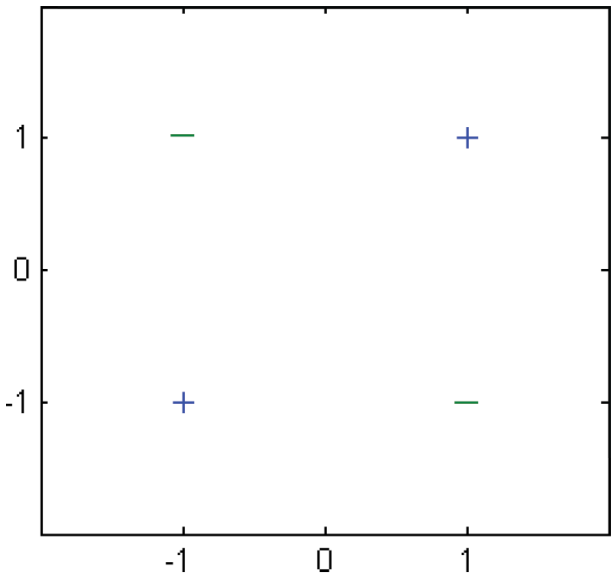
(3) Radial Basis Kernel : $K(x, x') = exp(-\frac{1}{2}||x - x'||^2)$

(d) What is the leave-one-out cross-validation error for the maximum margin based separation in the following figure ? (we are asking for a number) Please provide a (at least one-sentence) justification.



Question 2. Another Support Vector Machine

Consider a supervised learning problem in which the training examples are points in 2-dimensional space. The positive examples are $(1, 1)$ and $(-1, -1)$. The negative examples are $(1, -1)$ and $(-1, 1)$.



(a). (1 pts) Are the positive examples linearly separable from the negative examples in the original space?

(b). (4 pts) Consider the feature transformation $\phi(x) = [1, x_1, x_2, x_1x_2]$, where x_1 and x_2 are, respectively, the first and second coordinates of a generic example x. The prediction function is $y(x) = w^T * \phi(x)$ in this feature space. Give the coefficients , w, of a maximum-margin decision surface separating the positive examples from the negative examples. (You should be able to do this by inspection, without any significant computation.)

(c). (4 pts) What kernel $K(x, x')$ does this feature transformation ϕ correspond to?